**Week 2**

1. **Definition of Cloud Computing**

* On-demand availability of computer system resources, especially data storage, computing power, without direct active management by the user

1. **Essential cloud characteristic**

* On demand
* Self-serve
* Broadband network
* Rapid elasticity
* Metered Service Model

1. **Main provider and their market share**

* Amazon 60%
* Microsoft 30%
* Google 10%

1. **Why a company WOULD want to use cloud**

* Cost
  + Lower start-up cost
  + Lower than maintaining data center
  + Move cost from Capital to Operational
* Scalability
* Reliability
* Platform
* Accessibility
* Reliable

1. **Why company WOULD NOT want to user cloud**

* Different cost model
* Different IT skill set
* Privacy issues
* Data location
* Liability (tin tưởng)

1. **Why you want to learn about cloud**

* New platform to develop for
* Grate employment opportunity
* Opportunity to differentiate yourself

1. **Key technological innovation that enabled the creating of the cloud**

* Cheap fast hardware
* Fast broadband networks
* Hardware virtualization
* Open source software
  + Linux
  + Apache
  + MySQL
  + Cassandra
  + Redis
* High cost of an IT organization

1. **Cloud Service Model**

* **SaaS (Software-as-a-Service): End-User Application**
  + CRM, email, virtual desktop, communication, games, …
* **PaaS (Platform-as-a-Service)**
  + Execution runtime, database, webserver, development tools, …
* **IaaS (Infrastructure-as-a-Service)**
  + Virtual machine, servers, storage, load balancers, network, …

1. **Cloud Delivery Model**

* Public cloud: services offered by public provider
* Private cloud: services offered in-house corporate data center
* Hybrid could: offered both in-house and by cloud provider
* **Hybrid Cloud is the most popular model (94% of respondents using)**

**==============================================================================**

**Week 3**

1. **Core AWS service offerings**

* EC2 (Elastic Compute Cloud): Highly managed VM
* S3 (Simple Storage Service): Provide storage
* IAM (Identity and Access Management): Security configuration
* Lambda: Part of AWS serverless structure
* RDS (Relational Database Service): Provide database that AWS will manage for you
* SQS (Simple Queue Service)
* CloudWatch: Data collection, monitoring, management services
* API Gateway: Let developer create AWS-host API
* ELB (Elastic Load Balancer): Distribute incoming requests
* Route53: AWS’s DNS
* VPC: Virtual networking environment

1. **AWS is**

* Global Cloud Computing platform
* Currently the largest provider
* Provides IaaS and PaaS
* Customer host SaaS on its platform

1. **Different between Region and Availability Zone**

* Region
  + Separate geographic areas that AWS uses to house its infrastructure
  + Current 23 of them
  + Not all regions have all AWS services
  + Different region has different cost
* Availability Zone (AZ)
  + Fault-independent data centers within a region
  + Redundant everything (power, network, cooling, …)
  + High-speed connectivity
  + Most services support replication and fail-over other AZ
  + Each region has at least **2** **AZ**

1. **Share Responsibility Model**

* Amazon responsible for
  + Compute
  + Storage
  + Database
  + Networking
  + AZ, regions, edge location
* Users responsible for
  + Customer Data
  + Applications
  + Platform
  + Operating System & Network Configuration
  + Data Encryption
  + Networking Security

1. **Amazon Free Tier**

* Free for a year
* 720 hours of most things

1. **Global Reach**

* Ability to deploy application anywhere in the world

1. **Different between Reliability, Availability, Durability**

* Reliability (minh bạch)
  + A user may forget about the other issues of their data being secure and available in the cloud without having to be at risk personal loss
* Durability
  + The tiny errors that occur in files that could make on or more individual bytes get corrupted or lost when write, read, rewrite
  + 11 9’s durable
* Availability
  + The percentage of time an application and its services are available
  + 6 9’s availability

1. **Fail-Over**

* A system usually used for mission -critical programs or reusable services that can introduce a single point of failure for multiple applications
* Can span more than one geographical region

**==============================================================================**

**Week 4**

1. **IAM (Identity and Access Management) security model**

* Include
  + Users
  + Groups
  + Policies
* Principle of least privilege
  + Super important
  + Allow user to do what they need and no more
  + **Best practice:** review users’ permission to ensure this permission is adhere to

1. **Authentication vs Authorization**

* Authentication: establish who you are
* Authorization: establish if you are allowed to do an action

1. **Security best practice**

* Turn on MFA
* **Disable root access**
* Rotate credential
* Don’t share credentials
* Create admin user and groups
* Use group as much as possible
* Don’t apply direct permission to users
* Audit who has access to what
* User policy condition

1. **Amazon Resource Name (ARN)**

* Used in policies to identify resources the policies apply to

1. **Multifactor Factor Authentication (MFA)**

* 2 step verification
* Managed under security credentials

**==============================================================================**

**Week 5**

1. **Serverless key features**

* Event driven
* Code focused
* No infrastructure management
* **Use serverless as much as possible**

1. **AWS Serverless services**

* Lambda
* API Gateway
* S3
* AuroraDB serverless
* SQS
* CloudWatch
* CodeCommit

1. **Event-Driven Programming**

* Code that handle event of the application
* Event fires -> code execute

1. **Constraints in Lambda**

* 1000 current functions
* 250MB uncompressed max package size
* 15 mins max execution time
* 3GB RAM (configurable) – **Charge by GB/min**
* 6MB invocation payload (smaller if async)
* **Can easily work arounds by breaking up large function**

1. **REST Endpoint**

* A web service
* Build on top of HTTP
* Takes a **URL**, **query string parameters** (optional), a **method type**, **request body** (optional) then **executes** an **operation** and **returns** a **response** code and a **result**

1. **Connection between API Gateway and Lambda function**

* Good way to integrate different platform
* Scaling, monitoring, fail-over, security is all integrated
* No infrastructure to manage
* API cache included
* API versioning
* Canary releases
* Faster time to market

**==============================================================================**

**Week 6**

Review Questions

1. **EC2 User Data is used for**
2. Identity
3. **Configuration**
4. Authorization
5. Logging
6. **Access time to DynamoDB table is dependent on the table size**
7. True
8. **False**
9. **A DynamoDB table’s sort key is required**
10. True
11. **False**
12. **Horizontal scaling is better than vertical scaling**
13. **True**
14. False
15. **Lambda event handlers can be synchronous or asynchronous**
16. **True**
17. False
18. **RDS Snapshots are stored in EFS**
19. True
20. **False**
21. **Which tool is used to help create IAM policies**
22. VS Code AWS extension
23. AWS Policy Fabricator
24. AWS Command Line
25. **AWS Policy Generator**
26. **If I had a legacy app running in my data center and I wanted to add scaling, I could**
27. Re-host it
28. Re-factor it
29. **Re-platform it**
30. Re-invent it
31. **In AWS Lambda there is no infrastructure to manage**
32. **True**
33. False
34. **DES encryption is stronger than AES encryption**
35. True
36. **False**
37. **The google authenticator is used with AWS MFA**
38. **True**
39. False
40. **The kind of backup used by RDS**
41. Partial
42. Binary
43. **Incremental**
44. Quick
45. **Compression can improve an application’s performance**
46. **True**
47. False
48. **ECS boot volumes can be on**
49. S3
50. EFS
51. **EBS**
52. **Instance storage**
53. Instance Storage can be snap-shot
54. True
55. **False**
56. **EC2 instances can be reserved for only 8 hours a day for 3 years**
57. **True**
58. False
59. **When provisioning an EC2 instance, is best to**
60. **Start small and scale up**
61. Start large and scale down
62. Start wide and scale in
63. **Start narrow and scale out**
64. **The maximum RAM of Lambda function is**
65. 128MB
66. 1GB
67. **3GB**
68. 4GB
69. **IP Addresses can be scarce resource in an auto-scaled application**
70. **True**
71. False
72. **Which is not associated with RDS**
73. Read Replicas
74. Security groups
75. **ACL**
76. Snap shots
77. **Availability zones are**
78. Fault-dependent within a region
79. Co-located within a region
80. **Fault-independent within a region**
81. Not found within regions
82. **The biggest reason companies adopt the cloud is faster access to infrastructure**
83. **True**
84. Fault
85. **VSphere is an example of the public cloud**
86. True
87. **False**
88. Where is the best place to get AWS training after ICS200 is complete
89. Amazon
90. Pluralsight
91. **A cloud guru**
92. None of the above
93. **Which is not associated with EC2**
94. AMI
95. Security group
96. **S3**
97. EBS
98. **Which is most like a firewall**
99. IAM policy
100. **EC2 security group**
101. EC2 autoscaling group
102. VPC
103. **Netflix is so good with the cloud, it only has to employ**
104. Less than 10 operations engineers
105. Less than 1000 operations engineers
106. **Less than 100 operations engineers**
107. Less than 5000 operations engineers
108. **API Gateway is an example of**
109. IaaS
110. FaaS
111. SaaS
112. **PaaS**
113. **A reason to not use the cloud for you capstone would be**
114. Vendor lock-in
115. Lack of expertise
116. Security concerns
117. **Stakeholder refusal**
118. **AWS Services use**
119. Javascript user the hood
120. **TCP/IP under the hood**
121. UDP under the hood
122. Web sockets under the hood
123. **There are 2 ways to secure an S3 bucket**
124. **True**
125. False
126. **API Gateway can be used to**
127. **Create Rest Endpoints on Lambda**
128. **Create Rest Endpoints on Apache**
129. **Create APIs on Web Sockets**
130. Create on raw TCP
131. **EFS has strong consistency**
132. **True**
133. Fault
134. **Which is not a security best practice**
135. Use groups to manage permissions
136. **Practice principle of shared responsibilities**
137. Rotate credentials
138. Audit the credentials report
139. **The best way to debug a Lambda function would be to**
140. Use the AWS Management Console debugger
141. **Use console.log statements**
142. Use VS Code remote debugging
143. **Develop it locally first then deploy it**
144. **You can write a Lambda function in C++**
145. True
146. **Fault**
147. **DynamoDB has 11 9’s of durability**
148. True
149. **Fault**
150. **If you are using off-the-shelf AWS components like cognito or IAM there is no need to verify security**
151. True
152. **Fault**
153. **Which is not part of the API Gateway**
154. Throttling
155. HTTP Proxy Integration
156. **Security groups**
157. IAM Integration
158. **An identity provider is used for authorization**
159. True
160. **Fault**
161. **Which is not part of an IAM policy**
162. Condition
163. Effect
164. **ARN**
165. Action
166. **DynamoDB Items have ARNs**
167. True
168. **Fault**
169. **Durability speaks to resource being there when you need it**
170. True
171. **Fault**
172. **One thing not associated with DynamoDB is**
173. Primary key
174. Partition key
175. Transactions
176. **Relations**
177. **A reason not to use AWS Lambda would be**
178. It can be expensive
179. **The underlying application may not be a good fit**
180. **There can be performance issues**
181. **Vender lock-in is a concern**
182. **Which is not an encryption algorithm**
183. RSA
184. **Public key**
185. AES
186. DES
187. **Eventual consistency means if no new updates are made eventually all accesses will return the last value written**
188. **True**
189. Fault
190. **EC2 instances are great for legacy enterprise apps**
191. **True**
192. Fault
193. **With Lambda you don’t get**
194. Charged by the GB/S
195. Automatic Failover
196. Automatic scaling
197. **Automatic OS Updates**
198. **It’s easiest to scale RDS**
199. **Vertically**
200. Horizontally
201. **Manually**
202. Automatically
203. **Security can be traded for cost**
204. True
205. **Fault**
206. **A Lambda Even Handler**
207. Handles its own authentication in code
208. **Can invoke other Lambda Functions**
209. **Contains 3 parameters, only one of which is required**
210. Is best written in the AWS Management Console
211. **MFA is based on something you have and something you know**
212. **True**
213. Fault
214. **EFS can be unreliable**
215. True
216. **Fault**
217. **There is an upper limit to the number of objects in an S3 bucket**
218. True
219. **Fault**
220. **Reliability can be traded for cost**
221. **True**
222. Fault
223. **S3 is an example of a(n)**
224. NoSQL database
225. **Key-value store**
226. Block store
227. Relational stone